Proceedings of Machine Learning Research 230:1–23, 2024 Conformal and Probabilistic Prediction with Applications

# CoPAL: Conformal Prediction for Active Learning with Application to Remaining Useful Life Estimation in Predictive Maintenance

**Zahra Kharazian** ZAHRA.KHARAZIAN@DSV.SU.SE
**Tony Lindgren** TONY@DSV.SU.SE
**Sindri Magnússon** SINDRI.MAGNUSSON@DSV.SU.SE
*Department of Computer and Systems Sciences (DSV), Stockholm University, Sweden*

**Henrik Boström** BOSTROMH@KTH.SE
*School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden*

**Editor:** Simone Vantini, Matteo Fontana, Aldo Solari, Henrik Boström and Lars Carlsson

## Abstract

Active learning has received considerable attention as an approach to obtain high predictive performance while minimizing the labeling effort. A central component of the active learning framework concerns the selection of objects for labeling, which are used for iteratively updating the underlying model. In this work, an algorithm called *CoPAL* (Conformal Prediction for Active Learning) is proposed, which makes the selection of objects within active learning based on the uncertainty as quantified by conformal prediction. The efficacy of CoPAL is investigated by considering the task of estimating the remaining useful life (RUL) of assets in the domain of predictive maintenance (PdM). Experimental results are presented, encompassing diverse setups, including different models, sample selection criteria, conformal predictors, and datasets, using root mean squared error (RMSE) as the primary evaluation metric while also reporting prediction interval sizes over the iterations. The comprehensive analysis confirms the positive effect of using CoPAL for improving predictive performance.

**Keywords:** Conformal Prediction, Active Learning, Machine Learning, Regression, Predictive Maintenance, Remaining Useful Life prediction, and Time Series.

## 1. Introduction

In decision-making processes, humans have the ability to adapt and optimize their decisions based on new information, allowing them to refine their decisions over time iteratively. In contrast, machine learning algorithms typically do not have this flexibility; their decision is based on the dataset on which they are trained. This inherent limitation of machine learning highlights the need to include human intervention and supervision in machine learning decision-making processes. Human-in-the-loop approaches provide the opportunity for machine learning models to enhance their decisions under the supervision of humans. Active learning is a methodology that employs the human-in-the-loop strategy in machine learning (Settles, 2009). It refines the model's predictive efficiency by retraining iteratively and incorporating the feedback loops from human experts. It is also known for its power to optimize the labeling process through strategic sample selection and annotation, especially in scenarios where labeled data is limited or costly to obtain. Active learning works on the assumption that not all of the samples in the dataset are equally informative for model improvement. To select the most informative ones for annotation or label correction, there are various query strategies. Among all, *uncertainty-based query strategies* are significant for their ability to effectively use the uncertainty level in the model's prediction for selecting and querying informative samples to oracle in active learning. To quantify the uncertainty level in predictions, many approaches have been considered (Settles, 2009), e.g., Shannon's Entropy (Shannon, 1948), decision trees (Lewis and Catlett, 1994), and nearest-neighbor classifiers (Fujii et al., 1999). More recently, *Conformal Prediction* has emerged as a highly beneficial approach for obtaining uncertainty scores for data samples. Conformal prediction is particularly useful as it offers reliable uncertainty measures in prediction tasks instead of point predictions. While conformal prediction has been widely utilized for different tasks, only a few studies investigate the effect of using techniques from this area for active learning, e.g., (Matiz and Barner, 2019, 2020). However, these studies considered active learning for classification tasks. To the best of our knowledge, this work is the first to investigate the benefits of using conformal prediction for active learning on regression tasks. The main contributions of this paper are summarized as follows:

- We propose the CoPAL algorithm, the first model-agnostic algorithm that integrates Conformal Prediction in the Active Learning framework for regression tasks.
- We demonstrate the efficacy of our CoPAL algorithm in enhancing performance in time-series regression problems.
- We illustrate the benefits of the CoPAL algorithm on Predictive Maintenance problems using real-world data.

One specific use case of this algorithm is enhancing the Remaining Useful Life (RUL) estimation in Predictive Maintenance (PdM). Original Equipment Manufacturers (OEMs) and vehicular companies aim to implement an optimized maintenance strategy to enhance operational efficiency and reduce operational costs for their customers. Predicting the RUL of components using PdM techniques emerges as a crucial solution to achieving this objective. However, the effectiveness of RUL prediction relies heavily on access to a large volume of data with corresponding labels and true RUL, which may not always be readily available in real-world applications. Moreover, the practicality of sending all vehicles to workshops for RUL estimation is limited due to logistical constraints and resource considerations. Conducting comprehensive checks on every vehicle within a large fleet is time-consuming and resource-intensive. Furthermore, not all vehicles may require immediate attention or extensive maintenance interventions. In such scenarios, there is a clear need to implement a more strategic approach for selecting vehicles for RUL estimation that optimizes resource utilization and maximizes the effectiveness of predictive maintenance efforts. This research uses the proposed CoPAL methodology to tackle such challenges in PdM since it is based on active learning and conformal prediction.

The subsequent sections of this paper are organized as follows: Section 2 reviews work related to this study. Section 3 presents the problem formulation and describes CoPAL. The use case of this algorithm on a time series dataset in the PdM domain is provided in Section 4. Experiments and results are demonstrated in Section 5. Finally, in Section 6, the conclusion of this study is discussed together with possible directions for future work.

## 2. Background

### 2.1. Active Learning

Active learning is a paradigm in machine learning that iteratively selects the most informative samples from an unlabeled pool of data and sends them as a query to the oracle (a supervisor or an expert) to label or correct the label of the selected samples with the goal of improving the model's performance by experiencing and retraining (Settles, 2009, 2012). One of the important aspects that underpin active learning is having a model-agnostic nature. This principle creates flexibility in freely choosing the desired model. One application of active learning is the efficient use of labeling resources. This paradigm reduces the need for manually labeling the whole dataset. Instead, it focuses on selecting the most informative samples. This is beneficial in terms of saving time and resources, especially when labeling large datasets. Some examples of this application can be seen in (Kharazian et al., 2023; Barata et al., 2021; Chen and Mani, 2011; Gissin and Shalev-Shwartz, 2019), where active learning helped annotate a highly imbalanced and large dataset more efficiently and refined the model's performance. In general, an active learning strategy can be summarized in the following steps: 1) Initialization: Selecting a small labeled training set and a large unlabeled pool. 2) Model training: Training a naive model on the selected samples. 3) Sample selection: Predicting the pool instances' labels using the trained model and selecting the most informative samples based on the desired query strategy. 4) Feedback: Label or correct the selected samples' labels. 5) Model update: re-train the model with newly added samples.

In this paradigm, various sample selection strategies, a.k.a, *query strategies*, can be used to select the most informative samples according to the desired application. *Uncertainty sampling* (Lewis, 1995), *Diversity Sampling*, and *Query-by-Committee Sampling* (Seung et al., 1992) are among the existing common query strategies. *Uncertainty sampling* is one of the most popular methods that select the samples based on their uncertainty scores, which measures the model's uncertainty or confidence in its predictions. Consequently, shows how reliable the model prediction for a given instance is. The higher the uncertainty score, the lower the confidence level in the prediction.

In this study, we estimate the remaining useful life of components, making the task regression-oriented. For this purpose, *uncertainty sampling* is selected as the query strategy, which selects samples based on their uncertainty score in prediction. In other words, the active learner queries samples with the model's confidence level in their prediction from the unlabelled pool.

### 2.2. Conformal Prediction

*Conformal prediction*, introduced in (Gammerman et al., 1998; Vovk et al., 2005; Papadopoulos et al., 2002) is a powerful framework that allows the error rate of any predictive model to be controlled. This is achieved by turning a point prediction into a prediction set with a guaranteed coverage rate; the user specifies a level of confidence, which gives a lower bound for the probability that the true target is included in the output prediction set. The framework is model-agnostic and can be used with any classification or regression model.

Since the coverage rate is guaranteed by the framework, under the assumption of exchangeability, conformal predictors are often evaluated with respect to the informativeness, e.g., the size of the prediction sets. The size of a prediction set may also give an indication of the uncertainty of the underlying model's point prediction. For regression problems, a larger (smaller) prediction interval indicates a higher (smaller) expected deviation between the predicted and actual value. It should be noted, however, that for standard (non-normalized) conformal regressors, all prediction intervals are of the same size, and they do hence not provide any specific information on the uncertainty of the point predictions.

To make these intervals more informative, normalization (Papadopoulos et al., 2008) may be employed by using some difficulty estimator. However, as pointed out in (Boström and Johansson, 2020), normalization has the drawback that the resulting prediction intervals may not reflect actual errors; the intervals may be many times larger (or smaller) than the largest (smallest) observed error. As a remedy, so-called Mondrian conformal regressors were proposed in (Boström and Johansson, 2020), by which a standard conformal regressor is formed for each Mondrian category, which in turn can be formed by, e.g., binning the difficulty estimates.

It should, however, be noted that for both Mondrian and normalized conformal regressors, the correlation between the size of the prediction interval and the (absolute) error of the prediction is determined by the employed difficulty estimator; for a non-informative (random) estimator, the correlation coefficient can be expected to be close to zero. This also holds for both normalized and Mondrian conformal predictive systems (Vovk et al., 2020; Boström et al., 2021), where the prediction intervals may be extracted from the output conformal predictive distributions.

### 2.3. Predictive maintenance (PdM)

Predictive maintenance (PdM) represents a transformative part of the industry as a whole (often referred to as Industry 4.0) for improving efficiency, and reliability, together with cost savings (Achouch et al., 2022; Revanur et al., 2020). PdM utilizes data analytics, machine learning algorithms, and Internet of Things (IoT) technologies to anticipate maintenance needs before they become critical issues. This proactive strategy is particularly important in the automotive sector, where vehicle performance and safety are of great importance. In the automotive industry, PdM is used for optimizing maintenance strategies (de Jonge and Scarf, 2020; Karlsson et al., 2023; Lindgren et al., 2013), minimizing downtime (Pavlopoulos et al., 2024; Rylander, 2023), and reducing repair costs (Biteus and Lindgren, 2017). The importance of maintenance planning will most probably increase even more with the advent of autonomous or semi-autonomous heavy vehicles. Here we refer to semi-autonomous vehicles as vehicles that have a driver that can interact with the vehicle if the autonomous driving system, for any reason, cannot cope with the traffic situation and hands over control to the human. In the near future, this would probably mean having a driver onboard a vehicle, but further ahead, human drivers could remotely take control of the truck(s), similar to drone operators of today. One key element here is that humans onboard a vehicle can act as very sensitive sensors, feel vibrations, smell exhaust gases, etc. This type of sensor feedback from onboard humans will not be available for autonomous vehicles, which need to be equipped with more sensors for handling autonomous driving but also for monitoring and predicting the health status of each vehicle. Thus, both introduce more complexity and the ability to create better data-driven PdM models. In preparation for such a scenario, there has been a fair bit of research for creating frameworks for handling both fully autonomous vehicles and semi-autonomous vehicles from a PdM perspective, for example, (Jones et al., 2024; Tao et al., 2022). Traditionally, maintenance strategies come in two main different types, see (Swanson, 2001): Corrective maintenance - where the asset runs until failure occurs, and Preventive maintenance - where parts of an asset are serviced or changed before failure. Both strategies often lead to inefficiencies, unexpected downtime, and increased costs. In contrast, predictive maintenance leverages statistical and data-driven insights and machine learning techniques to detect early signs of potential issues and predict asset failures before they occur preemptively, see (Rahat et al., 2020, 2022). This allows OEMs and industrial organizations to estimate future failures and schedule their maintenance activities at optimal times. The RUL prediction of components using regression models is a critical aspect of PdM that predicts the amount of time a component has left before it fails. The regression models in the PdM strategy can detect the time of failure by monitoring real-time equipment conditions and analyzing the historical data from that equipment, see (Rahat et al., 2023). Such data usually includes sensor readings, operating conditions, maintenance history, environmental variables, and component specifications.

## 3. Problem Formulation and Methodology

The overall workflow of the CoPAL approach is illustrated in Figure 1 and can be expressed by three layers: 1) Initialization, 2) Model training and sample selection, and 3) Model update. Algorithm 1 states the steps of CoPAL, and the subsequent section elaborates on these procedures in alignment with the objectives of this study.

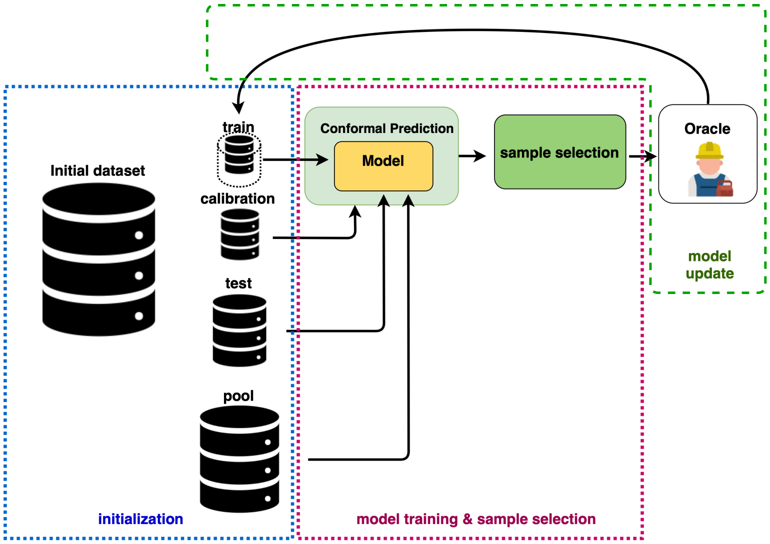


Figure 1: CoPAL workflow

**Algorithm 1: CoPAL** (Conformal Prediction for Active Learning)

```
   Input: $\mathcal{D}, model, n\_iterations, n\_rounds$
1  for $round \leftarrow 1$ to $n\_rounds$ do
       // Initialization
2      Separate $\mathcal{D}$ into $\mathcal{D}_{\text{L}}$ and $\mathcal{D}_{\text{U}}$;
3          $\mathcal{D}_{\text{L}} = \mathcal{D}_{\text{proper\_train}} \cup \mathcal{D}_{\text{calibration}}$;
4          $\mathcal{D}_{\text{U}} = \mathcal{D}_{\text{pool}} \cup \mathcal{D}_{\text{test}}$;
       // Model training & sample selection
5      for $iter \leftarrow 1$ to $n\_iterations$ do
6          $learner = model.fit(\mathcal{D}_{\text{proper\_train}})$;
7          $evaluate(learner, \mathcal{D}_{\text{test}})$;
8          for $i \leftarrow 1$ to $len(\mathcal{D}_{pool})$ do
9              $prediction\_interval \leftarrow$
                 $Conformal\_Prediction(learner, calibration, \mathcal{D}_{\text{pool}}[i])$;
10             $\mathcal{D}_{\text{pool}}[i].interval\_length =$
                 $prediction\_interval.max - prediction\_interval.min$;
11         end
12         $\mathcal{D}_{\text{pool}}.sort(by = interval\_length)$;
13         if $policy = most\_uncertain$ then
14             $selected\_samples \leftarrow select\_head(\mathcal{D}_{\text{pool}}, n\_samples)$
15         end
16         else if $policy = most\_certain$ then
17             $selected\_samples \leftarrow select\_tail(\mathcal{D}_{\text{pool}}, n\_samples)$
18         end
19         else if $policy = random$ then
20             $selected\_samples \leftarrow select\_random(\mathcal{D}_{\text{pool}}, n\_samples)$;
21         end
           // Model Update
22         $corrected\_selected\_samples \leftarrow Query(selected\_samples)$;
23         $\mathcal{D}_{\text{pool}} \leftarrow \mathcal{D}_{\text{pool}} \setminus selected\_samples$;
24         $\mathcal{D}_{\text{proper\_train}} \leftarrow \mathcal{D}_{\text{proper\_train}} \cup corrected\_selected\_samples$;
25     end
26 end
```

### 3.1. Initialization

Let $\mathcal{D}$ represent the available dataset. According to the definition of active learning, during the initialization layer, the CoPAL algorithm first separates the dataset into two parts (see also steps 2 to 4 of the Algorithm 1):

1. The labeled subset $\mathcal{D}_{\text{L}} = \{(X_i, y_i) : i \in I_{\text{L}}\}$ where $I_{\text{L}}$ is the set of indices corresponding to instances that have been labeled, and for each $X_i$, its corresponding label $y_i$ is known. This $\mathcal{D}_{\text{L}}$ is then split into two disjoint parts of $\mathcal{D}_{\text{proper\_train}}$ and $\mathcal{D}_{\text{calibration}}$.

2. The unlabeled subset $\mathcal{D}_\mathrm{U} = \{(X_i, \_) : i \in I_\mathrm{U}\}$, where $I_\mathrm{U}$ denotes the set of indices for which instances remain unlabeled, and hence, only the feature vectors $X_i$ are available without their corresponding labels. This $\mathcal{D}_\mathrm{U}$ is then split into two disjoint parts of $\mathcal{D}_\mathrm{pool}$ and $\mathcal{D}_\mathrm{test}$.

It is assumed that $\mathcal{D}$ constitutes a disjoint union of $\mathcal{D}_\mathrm{L}$ and $\mathcal{D}_\mathrm{U}$, formalized as $\mathcal{D} = \mathcal{D}_\mathrm{L} \cup \mathcal{D}_\mathrm{U}$ and $\mathcal{D}_\mathrm{L} \cap \mathcal{D}_\mathrm{U} = \emptyset$, ensuring that each instance in $\mathcal{D}$ is exclusively a member of either $\mathcal{D}_\mathrm{L}$ or $\mathcal{D}_\mathrm{U}$.

### 3.2. Model Training and Sample Selection

In the model training and sample selection layer (steps 5 to 21 of the Algorithm 1), a learner (initial model: $f$) is trained on the initial labeled proper_train set ($\mathcal{D}_\text{proper\_train}$) (step 6). Since CoPAL is a model-agnostic framework, any machine learning regressor model can be used for target prediction. In the next steps (8 to 11), a conformal predictor constructs the prediction intervals for the predicted target using a conformal prediction method like *Mondrian regressor* or a *conformal predictive system*. These conformal regressors use the trained model to make predictions on the calibration data ($\hat{y}_i = h\left(x_i\right)$ where $x_i \in \mathcal{D}_\mathrm{calibration}$). They also calculate the difficulty score of the samples in the calibration set using difficulty estimators. Then, the trained model is calibrated using the calibration set. Afterward, the difficulty score and the point prediction are also calculated for all samples in the pool set. Using this information, the model provides the prediction distribution of the pool set. We can get the prediction intervals from this distribution for the model's prediction depending on the desired confidence level. For instance, given 95% confidence, the prediction intervals are obtained from a CPS by setting the lower and higher percentiles to 2.5 and 97.5, respectively. The length of the predicted intervals for each sample is calculated, and these values are sorted in descending order in step 12. Now that these intervals are ready and sorted, in steps 13 to 21, the algorithm selects some instances (n_samples) using the desired query strategy $\mathcal{Q}$ or policy from $\mathcal{D}_\mathrm{pool}$ for labeling or label correction.

It is worth noting that the selection process hinges on a query strategy $\mathcal{Q}$, which evaluates the unlabeled instances' potential informativeness or value based on criteria such as uncertainty, representativeness, or a combination thereof. The selected samples would be different depending on the desired policy for sample selection. This study chooses five policies to evaluate their influence on model performance in predicting the samples' target. *Most_uncertain*, *most_certain*, and *random selection* are the main policies, and two more policies based on the *Roulette Wheel Selection* (a.k.a Fitness proportionate selection) from the first two main policies are considered for selecting samples. These policies are referred to as *Most_uncertain_roulette* and *Most_certain_roulette*. *Most_uncertain* policy selects the first n_samples (an arbitrary number) samples from the sorted pool having the highest prediction interval length. While the *Most_certain* policy selects n_samples with the smallest prediction interval length. The *random* policy is considered a baseline policy and selects random samples from the pool. Moreover, the two other policies (*Most_uncertain_roulette* and *Most_certain_roulette*) use a stochastic technique in genetic algorithms that selects samples for reproduction. Given a population, each sample gets a fitness score, and the roulette wheel selection calculates the total fitness of the population together with the probability of each sample being selected. This method simulates the spin of a roulette wheel, with each portion proportional to the samples' selection probability. This method ensures that the samples with higher fitness scores contribute more to the selection process while still allowing the selection of less informative samples to maintain the diversity of the population.

### 3.3. Model Update

After the sample selection step, the selected samples can be used to provide feedback to the model. According to Algorithm 1, in step 22, the selected samples are queried to oracle for checking and correcting the model's decision. Then, in steps 23 and 24, these samples will be removed from the $\mathcal{D}_{\text{pool}}$ and incorporated into the $\mathcal{D}_{\text{proper\_train}}$ with their correct labels, making the training set larger (This is showed with dotted dataset in Figure 1) and potentially more informative for the next model training. For the next iterations, all the steps from 5 to 25 will be repeated. In each iteration, the performance of the trained model will be evaluated on the test set (step 7). Finally, to have a stable evaluation of the CoPAL, the whole algorithm is repeated for n_rounds (here 5 rounds) based on n_rounds data splits and is evaluated by averaging the performance metrics across these rounds.

## 4. Application of CoPAL in Time Series Regression and Predictive Maintenance

This section illustrates and models one possible application of the CoPAL algorithm in enhancing the RUL prediction in PdM when the dataset is a multivariate time series.

Given $\mathcal{D}$ is a multivariate time series dataset. $X_t^v \in \mathbb{R}^N$ and $y_t^v$ are a feature vector and target variable (true RUL), respectively for vehicle $v$ at time $t$. Where $N$ is the number of covariants, $t \in T$ indicates the time step, and $v \in V$ represents the vehicle number.

This type of dataset includes temporal information, meaning that each vehicle has multiple readouts in its history until it fails. To include this temporal information in the model, we treat the temporal data from a vehicle as individual observations. Hence, each readout of a vehicle is considered an independent data point for that vehicle. Now the task is to train a predictive regressor model $f$ that accurately estimates the remaining useful life (RUL) of components using the multivariate time series data. Formally, we seek to learn a mapping $f : X \rightarrow Y$, where $X$ is the input space (multivariate time series feature space) and $Y$ is the output space (RUL values). For the sake of simplicity and without restricting the scope of our analysis, we can redefine each data point in the dataset as a pair consisting of input features $(X_i)$ and corresponding output or target value $(y_i)$, where $i$ represents the index of that specific data point within the dataset and $y_i = t_i^{lifetime}$ where:

$$t_i^{lifetime} = t_i^{failure} - t_i^{readout} \tag{1}$$

$t_i^{readout}$ and $t_i^{failure}$ representing the readout time and failure time for the $i^{th}$ sample, respectively.

Upon dataset preparation, it undergoes processing within Algorithm 1, as detailed earlier in Section 3. The only difference lies in incorporating the dataset's temporal aspect within this algorithm. In this case, we calculate the prediction intervals (steps 8 to 11) for all the readouts of each vehicle in the pool set. Finally, the average of the calculated intervals for each vehicle will be reported as the prediction interval of that vehicle. Considering the temporal information of the dataset in a real-world setting, allows the model to take into account the history of the vehicles when selecting informative vehicles as feedback to workshops for further assessment more robustly.

This approach also allows us to handle the inherent non-exchangeability of the time series data. In our study, individual samples were treated separately, and the prediction intervals were averaged for each vehicle to manage this non-exchangeability. Furthermore, it is crucial to keep observations from the same vehicles within the same split and not share them between different splits to avoid information leakage between data splits. By splitting the data based on vehicles rather than individual observations, we ensure that there is no such leakage and maintain the integrity of our model training and evaluation process. In other words, in this research, we have adopted vehicle-based exchangeability rather than sample-based exchangeability. This approach helps us better manage the inherent correlations within the time series data.

## 5. Experiments and Results

### 5.1. Experimental setup

This section delineates the experimental setup employed to investigate the performance of CoPAL in a regression task, specifically in RUL prediction. To achieve this goal, multiple experiments with different setups are designed. Table 1 summarizes the setup utilized in this study. These setups differ in the choice of machine learning models for training, conformal prediction technique for finding conformal intervals, following different policies for sample selection, the choice of the dataset, and finally, the evaluation method.

Table 1: Experimental setups

| Model | Query strategy | Conformal predictor | Data | Evaluation |
|---|---|---|---|---|
| XGBoost<br>Random Forest | Most-uncertain<br>Most-uncertain-roulette<br>Most-certain<br>Most-certain-roulette<br>Random | norm_Mond_CPS<br>std_Mond_CPS<br>Mond_reg | Component_X<br>C-MAPSS | RMSE<br>Test interval |

#### 5.1.1. Dataset

To assess the effect of the CoPAL framework on the regressor model's performance, two multivariate time series datasets in the field of PdM are selected; One real-world data from an engine component called Component_X of SCANIA trucks and one synthetic data from the turbofan jet engines from NASA. Both of these datasets are publically available, which makes this study reproducible.

**Component_X data** This real-world time series dataset is collected from an engine component from SCANIA trucks (Lindgren et al., 2024; Kharazian et al., 2024) and is available at the Swedish National Data Service [1]. It contains three sources of information about

[1] See https://doi.org/10.58141/1w9m-yz81

the Component_X of trucks. Operational data, time-to-event information, and specification of trucks. The operational data includes sensor readings collected from various onboard sensors producing more than one million readouts and 107 columns collected from around 23000 vehicles. Time-to-event data contains the time of the event, which is the time of repair of Component_X on the trucks for the first time. It also includes the label for the component that shows whether it has been repaired (repaired=1) or has not (repaired=0). Moreover, the time-to-event data is skewed toward the class 0. The specification data includes categorical variables that provide detailed specifications of the trucks. This research uses only the operational and time-to-event information to evaluate the CoPAL algorithm. To prepare this data for the experiments, we need to merge the information from the time-to-event data with the operational data. The true target (actual RUL) is calculated by subtracting the start time from the repair time according to Equation (1). Furthermore, for the RUL prediction task, only the readouts corresponding to repaired instances in the dataset are considered for analysis, while the healthy components are excluded. Finally, the rows with missing values are removed from the dataset, justified by their infrequent occurrence, composing less than 1% per variable.

**C-MAPSS data** The next dataset contains the operational and run-to-failure data from turbofan engines synthesized using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). This is also publicly available and contains four different sets of data with different experimental setups, such as different types of fault modes. For simplicity, we used the first set called *"train_FD001.txt"*, which considers only one type of failure. In total, it contains 21 numerical sensor measurements for each unit's operation time cycle. Moreover, this set includes 20631 readouts from 100 units, all of which have experienced failure, and each unit has multiple readouts for all of its operating cycles. The actual RUL for each unit is calculated using Equation (1) by subtracting the readout time from the last time_in_cycles for that unit in the dataset.

Thereupon, both datasets are split following the active learning paradigm. The initial percentage of samples in each set are as follows: proper_train (1%), calibration (14%), pool (60%), and test set (25%). Each set contains complete readouts of the selected unique vehicles.

### 5.1.2. Model

In this study, XGBoost (XGB) and Random Forest (RF) regressors with their default hyperparameters are chosen as training models due to their ability to handle complex datasets, high predictive accuracy, and faster training times.

### 5.1.3. Conformal Prediction method

For this study, three conformal predictors for regression tasks, including the *Mondrian regressor (Mond_reg)*, *normalized Mondrian conformal predictive system (norm_Mond_CPS)*, and *standard Mondrian conformal predictive system (std_Mond_CPS)* from the Crepes package (Boström, 2022) are used to obtain the prediction intervals and assess the uncertainty of the model in prediction. Furthermore, these conformal predictors typically involve hyperparameters like the confidence level, number of bins, and difficulty estimator that impact the conformal predictor's performance. In this study, since we focus on the order of the intervals' sizes for sample selection, changing the confidence level might not affect the result, so the confidence level is selected as the default value (95%). Also, the number of bins for Mond_reg, norm_Mond_CPS, and std_Mond_CPS is selected as 10, 5, and 5, respectively. Furthermore, the difficulty estimator from crepes.extra module is employed to estimate the difficulty of samples in the calibration set based on the standard deviation of the target variables of the k nearest neighbors (default k=25) in the training set.

#### 5.1.4. Query Strategies

Uncertainty sampling is chosen as the query strategy in the active learning setup of the CoPAL. In general, the choice of query strategy may vary depending on the application. Here, since estimating the RUL of components is of interest, Uncertainty sampling for the regression task is chosen to provide the uncertainty of the model's decision in prediction. Accordingly, five policies in sample selection are implemented to scrutinize the efficacy of CoPAL when choosing different samples to query the oracle for prediction correction. Most_Uncertain, Most_Uncertain_roulette, Most_Certain, Most_Certain_roulette, and Random selection are these policies in this study, which are explained in Section 3.2. Regarding choosing the number of vehicles to query the oracle in each iteration, 226 vehicles for Component_X and 9 units for the C-MAPSS dataset are considered in this study. These numbers are chosen according to several considerations: the total number of samples in the dataset, the number of iterations (here is 7), and the feasibility for expert review. Please note that the model was trained using the entirety of readouts obtained from each vehicle, encompassing multiple data points, rather than relying solely on individual readouts.

#### 5.1.5. Evaluation

To evaluate the performance of the regressor model in RUL prediction during iterations, the Root Mean Squared Error (RMSE) is selected. This metric measures the average of the residuals or errors between the predicted values and the actual values in each iteration of the CoPAL (see Equation (2)). In this study, using RMSE on the test set, the model's performance is assessed in every iteration of the active learning setup. The main benefit of choosing this metric is that the output of this measure has the same unit as the target variable (here, the time step, RUL). This makes it easy for the end user to interpret.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Another evaluation metric is to investigate the trend of the prediction intervals in the test set during the iterations. A decrease in interval length could be interpreted as the increasing power of the model through having more certainty in decision-making as the iterations progressed.

### 5.2. Experimental results

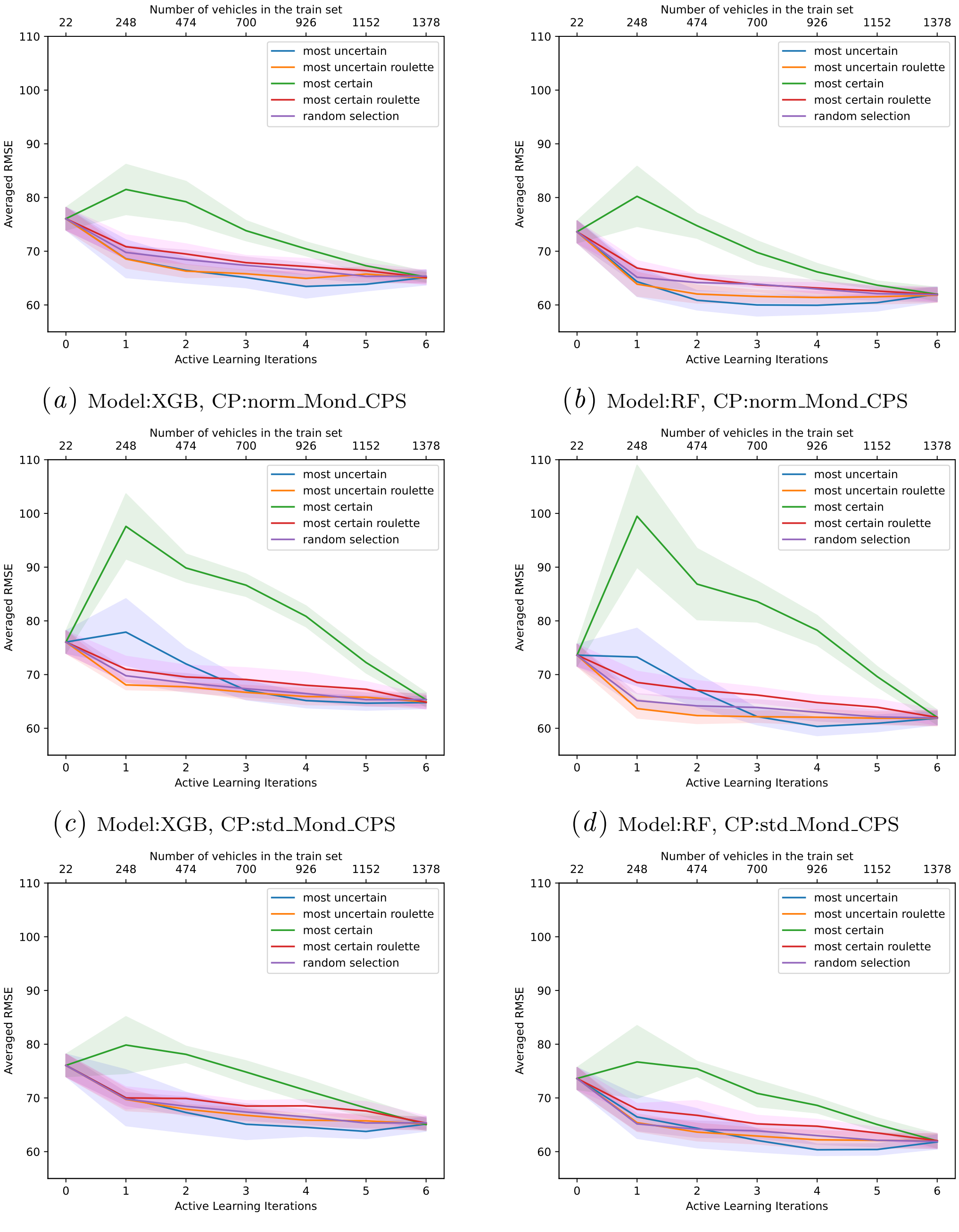
The goal of experiments in this study is to assess the performance of the CoPAL algorithm using different machine learning models, sample selection policies, conformal prediction methods, and datasets in predicting the Remaining Useful Life (RUL) of vehicle components. In general, we divide the experiments into two main parts: Experiments on the Component_X and experiments on the C-MAPSS dataset. For both datasets, two machine learning models, three conformal prediction methods, five sample selection policies, and two evaluation metrics are employed.

### 5.2.1. Experimental results on Component_X

Figure 2 illustrates the trend of RMSE in the CoPAL framework on the Component_X dataset. Figures 2(*a*), 2(*c*) and 2(*e*) demonstrate the performance of the XGB model over iterations for five policies in sample selection using norm_Mond_CPS, std_Mond_CPS, and Mond_reg, respectively. Furthermore, Figures 2(*b*), 2(*d*) and 2(*f*) make the same comparison of choosing different sample selection policies and different conformal prediction methods while having an RF model. Looking more closely, for instance, Figure 2(*b*) illustrates the RMSE trend in RUL prediction in 7 iterations of active learning. The x-axis shows the iteration number at the bottom of the figure and the number of vehicles used for training in that iteration at the top of the figure (Please note that each vehicle includes multiple (between 5 to 303) readouts in different time steps). The continuous lines represent the average values, and shaded regions show the variance of RMSE in RUL prediction when the CoPAL is repeated 5 rounds with different data splits. Different colors correspond to the sample selection policy. See all these five rounds of CoPAL before averaging for Figure 2(*b*) in Figure 7 in the Appendix A. Referring to Figure 2, generally, the error level decreased for both models and all scenarios after the desired model was retrained iteratively. Moreover, it is evident that two *most uncertain* and *most uncertain roulette* sample selection policies mostly outperform the random and other policies, especially when norm_Mond_CPS is selected for calculating the prediction intervals. While the most certain and most certain roulette policies showed less contribution to enhance the performance.

A more detailed analysis of the results can be found in Table 2. This table compares the performance of XGB and RF using the top two policies (most uncertain and most uncertain roulette) and random policy when choosing different conformal predictors. Overall, the error level decreased for both models and all scenarios after the desired model was retrained iteratively. Please note that each numeric cell denotes the mean and variance of RMSE for 5 different test sets (derived from five data splits) repeated in the corresponding iteration. The best result derived by using each model, is reported in bold and Italics. More specifically, the RF model could achieve the best results in RUL prediction (59.93 ± 1.6) for the Component_X dataset using *norm_Mond_CPS* for prediction interval calculation and the *most uncertain* sample selection policy. The second-best result of the RF model (60.34 ± 1.0) showed in bold comes with the *std_Mond_CPS* and *Mond_reg* when choosing *the most uncertain* policy. We can also conclude that using RF and the best sample selection policy, on average, the RMSE has decreased 13.68 units from the first iteration (73.61) to the fifth iteration of active learning (59.93).

When choosing the XGB model, the best results can be achieved by using *norm_Mond_CPS* (63.44 ± 2.1) for prediction interval calculation and the *most uncertain* sample selection policy. And the second-best result (63.75 ± 1.3) is obtained through Mond_reg and the most_uncertain policy. This shows using the CoPAL algorithm, the RMSE decreased by 13.06 units from the first iteration of active learning (76.05) to the fifth iteration (63.44)



$(a)$ Model:XGB, CP:norm_Mond_CPS

$(b)$ Model:RF, CP:norm_Mond_CPS

$(c)$ Model:XGB, CP:std_Mond_CPS

$(d)$ Model:RF, CP:std_Mond_CPS

$(e)$ Model:XGB, CP:Mondrian-regressor

$(f)$ Model:RF, CP:Mondrian-regressor

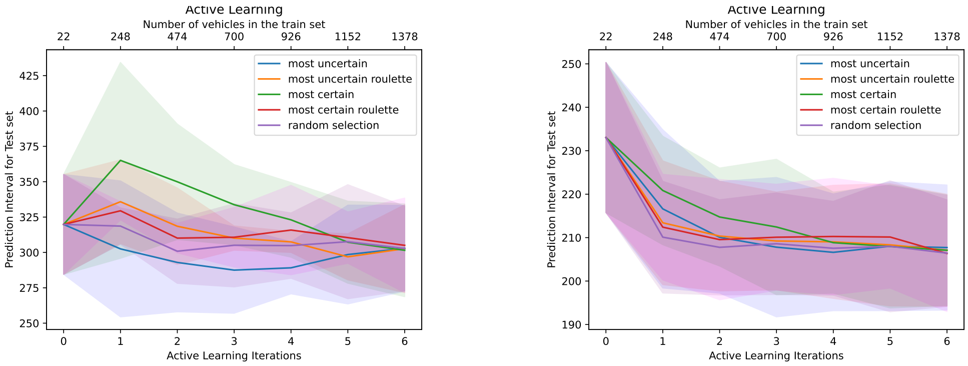
Figure 2: The RMSE trend on the Component_X dataset.

using the best setups for the Component_X dataset. In other words, on average, the model's predictions are now closer to the true remaining useful life values by approximately 13.06 time_steps.

Table 2: Data: Component_X, policy: all

| Model | itr | most uncertain | | | most uncertain roulette | | | random |
|---|---|---|---|---|---|---|---|---|
| | | Mond reg | norm-Mond-CPS | std-Mond-CPS | Mond reg | norm-Mond-CPS | std-Mond-CPS | any CPR |
| XGB | 0 | 76.05 ± 2.1 | 76.05 ± 2.1 | 76.05 ± 2.1 | 76.05 ± 2.1 | 76.05 ± 2.1 | 76.05 ± 2.1 | 76.05 ± 2.1 |
| | 1 | 70.03 ± 5.2 | 68.60 ± 3.5 | 77.90 ± 6.2 | 69.71 ± 2.0 | 68.56 ± 1.6 | 68.07 ± 0.9 | 69.78 ± 1.2 |
| | 2 | 67.28 ± 3.7 | 66.45 ± 2.3 | 71.99 ± 2.9 | 67.88 ± 0.8 | 66.30 ± 1.2 | 67.72 ± 0.7 | 68.44 ± 1.7 |
| | 3 | 65.09 ± 2.8 | 65.11 ± 1.9 | 67.09 ± 1.7 | 66.76 ± 1.3 | 65.80 ± 0.6 | 66.68 ± 1.3 | 67.37 ± 1.5 |
| | 4 | 64.53 ± 1.6 | ***63.44 ± 2.1*** | 65.15 ± 1.3 | 65.85 ± 1.0 | 64.93 ± 1.2 | 65.91 ± 1.5 | 66.46 ± 1.3 |
| | 5 | **63.75 ± 1.3** | 63.85 ± 1.2 | 64.66 ± 1.3 | 65.72 ± 1.2 | 65.74 ± 1.4 | 65.78 ± 1.5 | 65.33 ±1.2 |
| | 6 | 65.12 ± 1.3 | 65.12 ± 1.3 | 64.79 ± 1.1 | 65.14 ± 1.1 | 65.00 ± 0.9 | 64.90 ± 1.1 | 65.36 ± 1.1 |
| RF | 0 | 73.61 ± 2.0 | 73.61 ± 2.0 | 73.61 ± 2.0 | 73.61 ± 2.0 | 73.61 ± 2.0 | 73.61 ± 2.0 | 73.61 ± 2.0 |
| | 1 | 66.46 ± 4.0 | 64.33 ± 2.7 | 73.26 ± 5.3 | 65.39 ± 1.6 | 63.87 ± 2.2 | 63.67 ± 1.7 | 65.16 ±1.2 |
| | 2 | 64.35 ± 3.6 | 60.87 ± 1.8 | 67.10 ± 3.1 | 63.63 ± 1.6 | 62.03 ± 1.6 | 62.36 ± 1.4 | 64.17 ± 1.5 |
| | 3 | 62.08 ± 2.1 | 60.00 ± 2.0 | 62.17 ± 1.5 | 62.90 ± 1.4 | 61.59 ± 1.1 | 62.15 ± 1.0 | 63.87 ± 1.4 |
| | 4 | **60.34 ± 1.0** | ***59.93 ± 1.6*** | **60.34 ± 1.7** | 62.20 ± 1.7 | 61.39 ± 1.1 | 62.06 ± 1.4 | 62.98 ± 1.6 |
| | 5 | 60.40 ± 1.0 | 60.42 ± 1.5 | 60.92 ± 1.5 | 62.10 ± 1.4 | 61.53 ± 1.3 | 61.87 ± 1.1 | 62.11 ± 1.2 |
| | 6 | 61.80 ± 1.3 | 62.00 ± 1.3 | 61.87 ± 1.2 | 61.98 ± 1.3 | 61.82 ± 1.3 | 61.86 ± 1.4 | 61.90 ± 1.3 |

Ultimately, we devised a complementary experiment to assess the model's uncertainty in predicting the RUL of the test set and to monitor the length of decision intervals throughout the iterations of active learning. This was aimed at evaluating whether the intervals in the test set exhibited a reduction in length over successive iterations. A decrease in interval length would indicate that active learning facilitated the model in gaining greater certainty regarding its predictions for the test set as the iterations progressed. Here, we implemented this experiment on the Component_X dataset while having an RF model and norm_Mond_CPS and std_Mond_CPS conformal predictor.



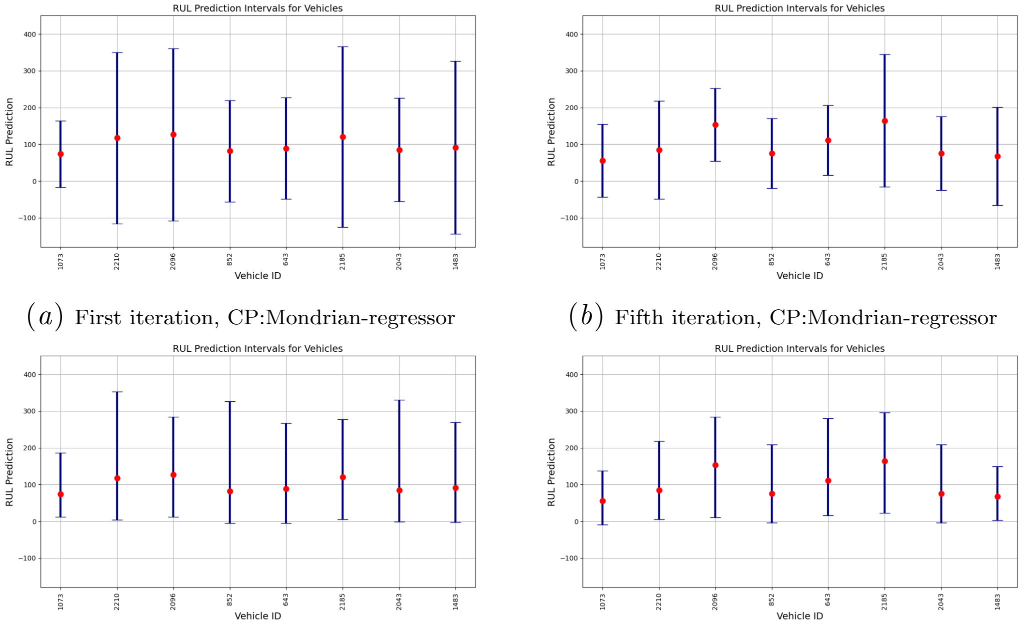
(*a*) norm_Mond_CPS

(*b*) std_Mond_CPS

Figure 3: The size of the prediction intervals for the test set of Component_X data in case the model is RF and the conformal predictor are std_Mond_CPS and std_Mond_CPS

The result of this complementary experiment shown in Figure 3 indicates a reduction in the length of prediction intervals for the test set and approves the improvement of the model's certainty in predicting the test set over iterations.

Delving deeper into the size of prediction intervals for the test set using CoPAL, Figure 4 depicts the prediction values and their corresponding intervals for a selection of random vehicles (for simplicity, one random readout from each random vehicle is chosen). The RF model is used in this experiment, along with Mond_reg and std_Mond_CPS as conformal predictors. Comparing Figure 4$(a)$ with Figure 4$(b)$ and Figure 4$(c)$ with Figure 4$(d)$, we can see that the prediction interval for these vehicles decreased over 5 iterations, meaning that the model became more certain in its decision using CoPAL algorithm.



$(a)$ First iteration, CP:Mondrian-regressor

$(b)$ Fifth iteration, CP:Mondrian-regressor

$(c)$ First iteration, CP:std_Mond_CPS

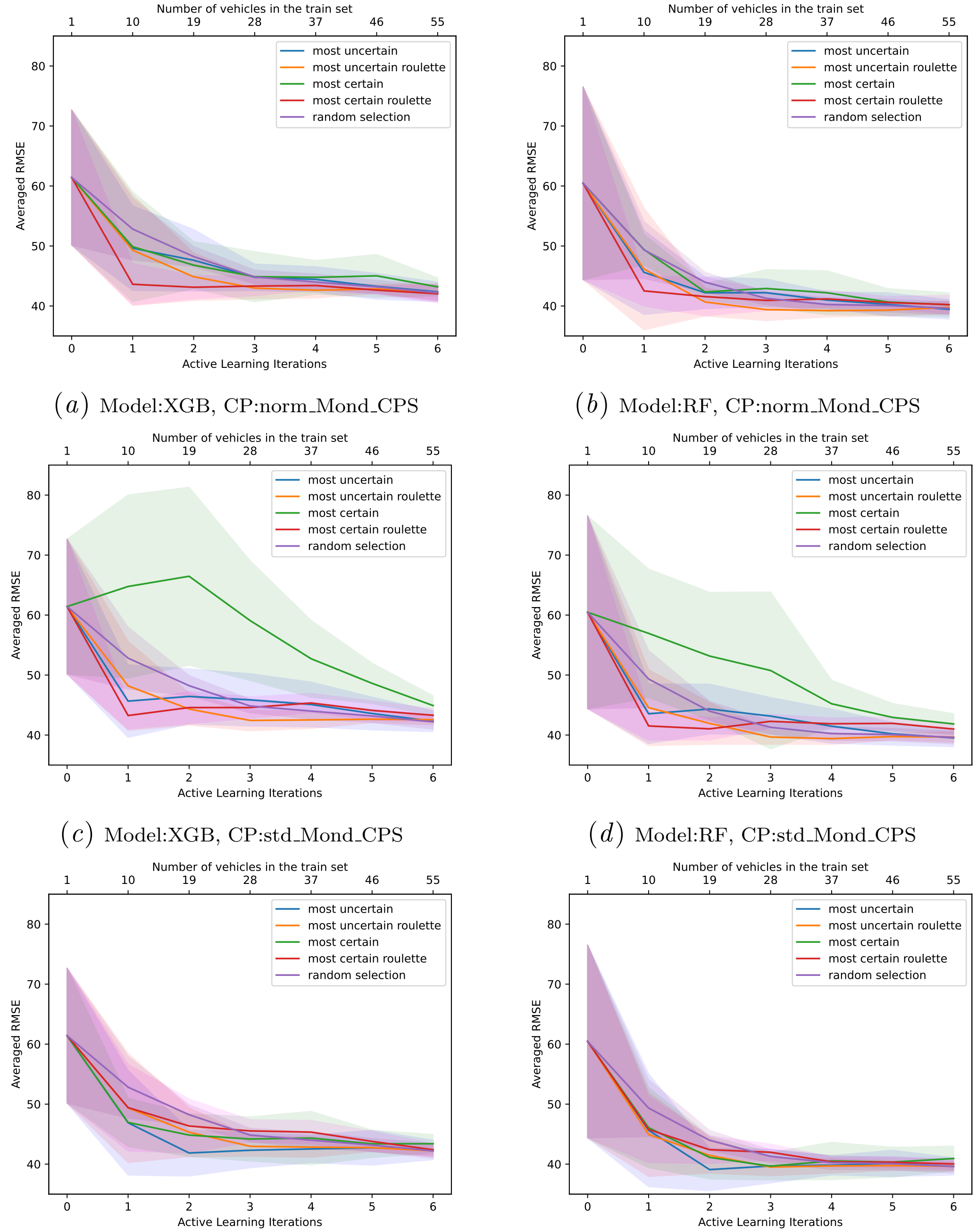
$(d)$ Fifth iteration, CP:std_Mond_CPS

Figure 4: The size of the prediction intervals for the test set of Component_X data in case the using RF as the model and Mondrian-regressor and std_Mond_CPS as conformal predictor

### 5.2.2. Experimental results on C-MAPSS

Figure 5 illustrates the averaged RMSE trend of the CoPAL framework on the C-MAPSS dataset during iterations over five rounds. In general, we can see a decreasing trend of averaged RMSE in all scenarios of the CoPAL algorithm.

Looking more closely, Figures 5$(a)$, 5$(c)$ and 5$(e)$ show the results for XGB model and Figures 5$(b)$, 5$(d)$ and 5$(f)$ illustrate the result for RF when having norm_Mond_CPS, std_Mond_CPS, and Mond_reg as conformal predictors, respectively. Here, in most cases, the most_uncertain, most_uncertain_roulette, and most_uncertain_roulette outperformed other policies. Furthermore, Table 3 contains the detailed analysis of CoPAL on the C-MAPSS dataset for the three most effective sample selection policies and random selection. The best result derived by using each model and its corresponding conformal predictor is reported in bold and italics. The XGB could achieve the best averaged-RMSE (41.85 ± 3.8) when



(*a*) Model:XGB, CP:norm_Mond_CPS

(*b*) Model:RF, CP:norm_Mond_CPS

(*c*) Model:XGB, CP:std_Mond_CPS

(*d*) Model:RF, CP:std_Mond_CPS

(*e*) Model:XGB, CP:Mond_reg

(*f*) Model:RF, CP:Mond_reg

Figure 5: The RMSE trend on the C-MAPSS dataset

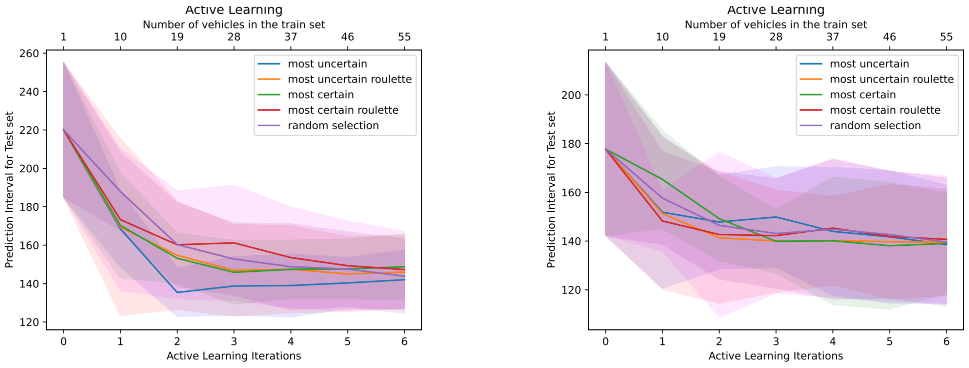
using Mond_reg and the *most uncertain* policy. The second-best result (42.02 ± 1.5) for this model happens using norm_Mond_CPS to select the most_certain_roulette samples selection policy. This shows using XGB in the CoPAL algorithm, the RMSE could be decreased by

19.56 units from the first iteration of active learning (61.41) to the third iteration (41.85) using the best setups for the C-MAPSS dataset. In the case of using RF as the model, the CoPAL algorithm achieves the best performance (39.08 ± 3.5) using Mond_reg to provide the prediction intervals using the most_uncertain sample selection policy. Moreover, the second best result (39.22 ± 1.0) is achieved by using norm_Mond_CPS and most_uncertain_roulette policies. Ultimately, we can conclude that the averaged RMSE decreased from 60.46 to 39.08 after three iterations using RF, resulting in 21.38 units better in RUL prediction.

Table 3: Data: C-MAPSS, policy: all

| Model | itr | most uncertain | | | most uncertain roulette | | | most certain roulette | | | random |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mond reg | norm Mond CPS | std Mond CPS | Mond reg | norm Mond CPS | std Mond CPS | Mond reg | norm Mond CPS | std Mond CPS | any CPR |
| XGB | 0 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 | 61.41 ± 11.2 |
| | 1 | 46.89 ± 8.7 | 49.63 ± 7.0 | 45.65 ± 6.0 | 49.36 ± 9.0 | 49.30 ± 9.1 | 48.19 ± 7.4 | 49.41 ± 7.2 | 43.61 ± 3.4 | 43.25 ± 2.2 | 52.82 ± 5.1 |
| | 2 | ***41.85* ± *3.8*** | 47.62 ± 5.1 | 46.43 ± 4.5 | 45.31 ± 4.0 | 44.86 ± 3.9 | 44.33 ± 2.6 | 46.35 ± 4.5 | 43.12 ± 1.9 | 44.58 ± 2.6 | 48.23 ± 1.6 |
| | 3 | 42.31 ± 2.9 | 44.79 ± 2.1 | 45.86 ± 4.3 | 42.97 ± 1.6 | 42.98 ± 1.7 | 42.43 ± 1.7 | 45.53 ± 1.8 | 43.31 ± 1.6 | 44.57 ± 1.7 | 44.84 ± 1.1 |
| | 4 | 42.52 ± 2.2 | 44.43 ± 2.1 | 45.09 ± 3.7 | 42.83 ± 1.6 | 42.62 ± 1.2 | 42.52 ± 1.4 | 45.33 ± 1.9 | 43.40 ± 0.6 | 45.34 ± 1.5 | 43.99 ± 1.3 |
| | 5 | 42.72 ± 2.8 | 43.27 ± 2.1 | 43.59 ± 2.7 | 42.74 ± 0.6 | 42.83 ± 0.7 | 42.63 ± 0.5 | 43.78 ± 1.6 | 42.64 ± 0.7 | 44.10 ± 1.5 | 43.18 ± 1.0 |
| | 6 | 42.37 ± 1.6 | 42.35 ± 1.5 | 42.30 ± 1.7 | 42.21 ± 0.8 | 42.24 ± 1.4 | 42.56 ± 0.8 | 42.39 ± 1.0 | **42.02 ± 1.5** | 43.29 ± 1.0 | 42.19 ± 1.2 |
| RF | 0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 | 60.46 ± 16.0 |
| | 1 | 45.63 ± 9.3 | 45.60 ± 7.0 | 43.54 ± 4.9 | 44.93 ± 6.9 | 46.15 ± 10.1 | 44.56 ± 6.3 | 45.76 ± 5.5 | 42.51 ± 2.5 | 41.51 ± 2.3 | 49.33 ± 4.7 |
| | 2 | ***39.08* ± *3.5*** | 42.20 ± 2.6 | 44.34 ± 4.1 | 41.45 ± 2.9 | 40.66 ± 2.3 | 41.94 ± 3.4 | 42.40 ± 2.2 | 41.56 ± 3.1 | 41.03 ± 1.8 | 43.97 ± 1.6 |
| | 3 | 39.68 ± 2.8 | 42.20 ± 2.2 | 43.15 ± 3.0 | 39.50 ± 1.4 | 39.38 ± 1.8 | 39.66 ± 1.1 | 41.97 ± 1.3 | 40.92 ± 1.6 | 42.28 ± 1.1 | 41.28 ± 1.0 |
| | 4 | 39.81 ± 1.6 | 40.98 ± 1.3 | 41.47 ± 2.8 | 39.69 ± 1.1 | **39.22 ± 1.0** | 39.40 ± 0.9 | 40.38 ± 0.7 | 41.17 ± 1.4 | 41.87 ± 0.9 | 40.24 ± 1.1 |
| | 5 | 40.12 ± 2.18 | 40.26 ± 1.9 | 40.20 ± 1.8 | 39.75 ± 0.7 | 39.30 ± 0.9 | 39.74 ± 0.5 | 40.31 ± 0.9 | 40.56 ± 1.0 | 41.94 ± 1.0 | 40.06 ± 1.0 |
| | 6 | 39.62 ± 1.4 | 39.43 ± 1.5 | 39.51 ± 1.4 | 39.64 ± 0.7 | 39.77 ±1.0 | 39.67 ± 0.8 | 40.03 ± 0.8 | 40.22 ± 1.5 | 40.99 ± 0.8 | 39.60 ± 1.0 |

Furthermore, similar to what we did for the Component_X dataset, here also, a complementary experiment is designed to assess the model's uncertainty in predicting the RUL of the test set by monitoring the decision interval's length over iterations. For this purpose, this experiment is implemented on the C-MAPSS dataset while having the setup for the two best results in Table 3. Figure 6 illustrates the decreasing size of the prediction intervals for the test set during iteration using RF as the model and both Mond_reg (Figure 6($a$)) and norm_Mond_CPS (Figure 6($b$)).



($a$) Mond_reg

($b$) norm_Mond_CPS

Figure 6: The size of the prediction intervals for the test set of C-MAPSS data in case the model is RF and the conformal predictor are Mond_reg and norm_Mond_CPS

## 6. Concluding Remarks

This study proposes CoPAL, an algorithm for active learning based on conformal prediction, to enhance the model's performance on regression tasks. To evaluate the effectiveness of this algorithm, we applied it to a real-world scenario focused on predictive maintenance, i.e., estimating the remaining useful life (RUL) of a vehicle component. The presented results show that the predictive performance can be significantly improved by using the CoPAL algorithm compared to random sample selection; the results on Component_X shows a reduction in the RMSE 13.06 and 13.68 units, respectively, for XGBoost and Random forests, which correspond to 16% and 17% improvement for these models. Furthermore, the results for the C-MAPSS dataset showed a 19.56 and 21.38 unit reduction of RMSE using XGBoost and Random forests, respectively, leading to 31% and 35% improvements.

In examining the result related to the effect of sample selection policies, it was observed that for selecting samples to query the oracle, utilizing the most_uncertain and most_uncertain_roulette policies proved notably advantageous for the Component_X data. However, in addition to these policies, the most_certain_roulette showed efficacy in refining models using the C-MAPSS dataset, especially when the CPSs are used for prediction interval calculation. This could be attributed to the smaller training sample size in the C-MAPSS dataset compared to the Component_X data, suggesting that the model benefits from exposure to more confidently predicted RUL values.

We then proceed to interpret the observed trends in the RUL prediction intervals for the test set over iterations. The results exhibited a reduction in the prediction interval's length over successive iterations on both datasets. This confirms the positive effect of the CoPAL algorithm in improving the model and gaining higher certainty in predicting the test set over iterations. Furthermore, To support the reproducibility of the experiments, the code is shared on a Gittea repository[2].

As a future work for this study, the CoPAL algorithm could be extended to incorporate survival analysis methodologies for examining the survival curve of vehicles and estimating their remaining useful life. This technique is particularly useful when dealing with real-world datasets where samples are often censored, meaning that they have not experienced failure or the event of interest in data collection. This enables the research to be done on the complete dataset rather than only on the samples that have experienced the failure. In this study, we utilized a separate calibration set to form the Mondrian categories and calibrate the model when using Mondrian conformal predictors. Future research could investigate the use of CoPAL without the calibration step. Additionally, future work could develop techniques to use fresh calibration instances on each iteration in an active learning setup. Exploring different query strategy methods beyond uncertainty-based sampling in active learning is also of interest. Lastly, applying the CoPAL algorithm to domains other than predictive maintenance, such as the medical field, could be a valuable direction for future research.

[2] See https://gitea.dsv.su.se/zakh1874/CoPAL/src/branch/main/

## Acknowledgments

This work has been funded by Scania CV AB and the Vinnova program for Strategic Vehicle Research and Innovation (FFI) through the project RAPIDS (grant no. 2021-02522).

## References

Mounia Achouch, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim, and Mehdi Adda. On predictive maintenance in industry 4.0: Overview, models, and challenges. *Applied Sciences*, 12(16), 2022. ISSN 2076-3417. doi: 10.3390/app12168081. URL https://www.mdpi.com/2076-3417/12/16/8081.

Ricardo Barata, Miguel Leite, Ricardo Pacheco, Marco OP Sampaio, João Tiago Ascensão, and Pedro Bizarro. Active learning for imbalanced data under cold start. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

Jonas Biteus and Tony Lindgren. Planning flexible maintenance for heavy trucks using machine learning models, constraint programming, and route optimization. *SAE International Journal of Materials and Manufacturing*, 10(3):306–315, 2017.

Henrik Boström. crepes: a python package for generating conformal regressors and predictive systems. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson, editors, *Proceedings of the Eleventh Symposium on Conformal and Probabilistic Prediction and Applications*, volume 179 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Henrik Boström and Ulf Johansson. Mondrian conformal regressors. In *Conformal and Probabilistic Prediction and Applications*, pages 114–133. PMLR, 2020.

Henrik Boström, Ulf Johansson, and Tuwe Löfström. Mondrian conformal predictive distributions. In *Conformal and Probabilistic Prediction and Applications*, pages 24–38. PMLR, 2021.

Yukun Chen and Subramani Mani. Active learning for unbalanced data in the challenge with multiple models and biasing. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 113–126. JMLR Workshop and Conference Proceedings, 2011.

Bram de Jonge and Philip A. Scarf. A review on maintenance optimization. *European Journal of Operational Research*, 285(3):805–824, 2020. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2019.09.047. URL https://www.sciencedirect.com/science/article/pii/S0377221719308045.

Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga, and Hozumi Tanaka. Selective sampling for example-based word sense disambiguation. *arXiv preprint cs/9910020*, 1999.

Alexander Gammerman, Katy Azoury, and Vladimir Vapnik. Learning by transduction. pages 148–155, 01 1998.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Ryan Jones, Raj Bridgelall, and Denver Tolliver. Route risk index for autonomous trucks. *Applied Sciences*, 14(7), 2024. ISSN 2076-3417. doi: 10.3390/app14072892. URL https://www.mdpi.com/2076-3417/14/7/2892.

Nellie Karlsson, My Bengtsson, Mahmoud Rahat, and Peyman Sheikholharam Mashhadi. Baseline selection for integrated gradients in predictive maintenance of volvo trucks' turbocharger. In *VEHICULAR 2023-IARIA*, 2023.

Zahra Kharazian, Mahmoud Rahat, Fábio Gama, Peyman Sheikholharam Mashhadi, Sławomir Nowaczyk, Tony Lindgren, and Sindri Magnússon. Aid4hai: Automatic idea detection for healthcare-associated infections from twitter, a framework based on active learning and transfer learning. In *International Symposium on Intelligent Data Analysis*, pages 195–207. Springer, 2023.

Zahra Kharazian, Tony Lindgren, Sindri Magnússon, Olof Steinert, and Oskar Andersson Reyna. Scania component x dataset: A real-world multivariate time series dataset for predictive maintenance, 2024.

David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pages 13–19. ACM New York, NY, USA, 1995.

David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

Tony Lindgren, Håkan Warnquist, and Martin Eineborg. Improving the maintenance planning of heavy trucks using constraint programming. In *ModRef 2013: The Twelfth International Workshop on Constraint Modelling and Reformulation, Uppsala, Sweden, September 16th, 2013*, pages 74–90. Université Laval, 2013.

Tony Lindgren, Olof Steinert, Oskar Andersson Reyna, Zahra Kharazian, and Sindri Magnússon. SCANIA Component X Dataset: A Real-World Multivariate Time Series Dataset for Predictive Maintenance, 2024. URL https://doi.org/10.58141/1w9m-yz81.

Sergio Matiz and Kenneth E Barner. Inductive conformal predictor for convolutional neural networks: Applications to active learning for image classification. *Pattern Recognition*, 90:172–182, 2019.

Sergio Matiz and Kenneth E Barner. Conformal prediction based active learning by linear regression optimization. *Neurocomputing*, 388:157–169, 2020.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*, pages 345–356. Springer, 2002.

Harris Papadopoulos, Alex Gammerman, and Volodya Vovk. Normalized nonconformity measures for regression conformal prediction. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2008)*, pages 64–69, 2008.

John Pavlopoulos, Alv Romell, Jacob Curman, Olof Steinert, Tony Lindgren, Markus Borg, and Korbinian Randl. Automotive fault nowcasting with machine learning and natural language processing. *Machine Learning*, 113(2):843–861, Feb 2024. ISSN 1573-0565. doi: 10.1007/s10994-023-06398-7. URL https://doi.org/10.1007/s10994-023-06398-7.

Mahmoud Rahat, Sepideh Pashami, Sławomir Nowaczyk, and Zahra Kharazian. Modeling turbocharger failures using markov process for predictive maintenance. In *30th European Safety and Reliability Conference (ESREL2020) & 15th Probabilistic Safety Assessment and Management Conference (PSAM15), Venice, Italy, 1-5 November, 2020*. European Safety and Reliability Association, 2020.

Mahmoud Rahat, Peyman Sheikholharam Mashhadi, Sławomir Nowaczyk, Thorsteinn Rognvaldsson, Atabak Taheri, and Ataollah Abbasi. Domain adaptation in predicting turbocharger failures using vehicle's sensor measurements. In *Phm society european conference*, volume 7, pages 432–439, 2022.

Mahmoud Rahat, Zahra Kharazian, Peyman Sheikholharam Mashhadi, Thorsteinn Rögnvaldsson, and Shamik Choudhury. Bridging the gap: A comparative analysis of regressive remaining useful life prediction and survival analysis methods for predictive maintenance. In *PHM Society Asia-Pacific Conference*, volume 4, 2023.

Vandan Revanur, Ayodeji Ayibiowu, Mahmoud Rahat, and Reza Khoshkangini. Embeddings based parallel stacked autoencoder approach for dimensionality reduction and predictive maintenance of vehicles. In *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning: Second International Workshop, IoT Streams 2020, and First International Workshop, ITEM 2020, Co-located with ECML/PKDD 2020, Ghent, Belgium, September 14-18, 2020, Revised Selected Papers 2*, pages 127–141. Springer, 2020.

Lina Rylander. Designing for change in complex systems : Design considerations for uptime in a transportation system with driverless vehicles, 2023.

Burr Settles. Active learning literature survey. 2009.

Burr Settles. Active learning. *Active Learning*, 2012. doi: 10.2200/S00429ED1V01Y201207AIM018.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

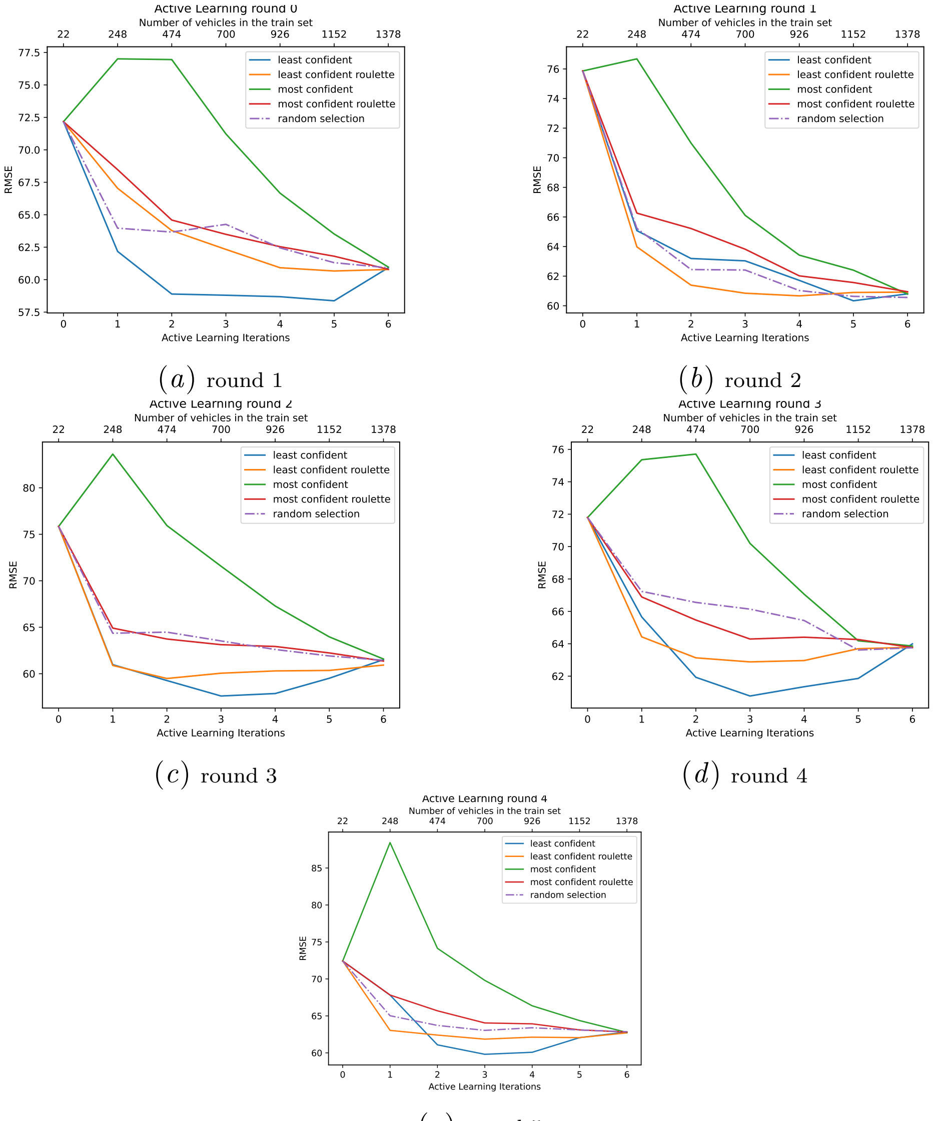
Laura Swanson. Linking maintenance strategies to performance. *International Journal of Production Economics*, 70(3):237–244, 2001. ISSN 0925-5273. doi: https://doi.org/10.1016/S0925-5273(00)00067-0. URL https://www.sciencedirect.com/science/article/pii/S0925527300000670.

Xin Tao, Jonas Mårtensson, Håkan Warnquist, and Anna Pernestål. Short-term maintenance planning of autonomous trucks for minimizing economic risk. *Reliability Engineering & System Safety*, 220:108251, 2022.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Valery Manokhin, and Alexander Gammerman. Computationally efficient versions of conformal predictive distributions. *Neurocomputing*, 397:292–308, 2020.

## Appendix A. First Appendix



$(a)$ round 1

$(b)$ round 2

$(c)$ round 3

$(d)$ round 4

$(e)$ round 5

Figure 7: Five rounds of CoPAL with RF and norm_Mond_CPS on Component_X dataset